INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2001; **35**: 247–280

Adaptive Q-tree Godunov-type scheme for shallow water equations

Ben Rogers[a], Masayuki Fujihara[b] and Alistair G. L. Borthwick[a,*]

[a] *Department of Engineering Science, University of Oxford, Parks Road, Oxford OX1 3PJ, U.K.*
[b] *College of Agriculture, Ehime University, 3-5-7 Tarumi, Matsuyama, 790-8566, Japan*

SUMMARY

This paper presents details of a second-order accurate, Godunov-type numerical model of the two-dimensional shallow water equations (SWEs) written in matrix form and discretized using finite volumes. Roe's flux function is used for the convection terms and a non-linear limiter is applied to prevent unwanted spurious oscillations. A new mathematical formulation is presented, which inherently balances flux gradient and source terms. It is, therefore, suitable for cases where the bathymetry is non-uniform, unlike other formulations given in the literature based on Roe's approximate Riemann solver. The model is based on hierarchical quadtree (Q-tree) grids, which adapt to inherent flow parameters, such as magnitude of the free surface gradient and depth-averaged vorticity. Validation tests include wind-induced circulation in a dish-shaped basin, two-dimensional frictionless rectangular and circular dam-breaks, an oblique hydraulic jump, and jet-forced flow in a circular reservoir. Copyright © 2001 John Wiley & Sons, Ltd.

KEY WORDS: adaptive quadtree grids; Roe's Riemann solver; shallow water equations

1. INTRODUCTION

Numerical models based on the non-linear shallow water equations (SWEs) are used to model predominantly horizontal, free surface flows in shallow lakes, wide rivers, estuaries and the coastal zone (in which case the SWEs are coupled through radiation stresses to a wave climate model). Natural flow domains have complicated boundary configurations, which may themselves strongly influence the interior flow patterns. The interior domain may contain regions of high-flow gradients, such as the shear layers in the vicinity of eddies, as well as regions of low hydrodynamic activity (stagnation zones). It is important for numerical shallow-flow models to be able to simulate accurately the evolution of relatively small-scale features, such as fronts, within large complex domains. To this end, research effort has been directed towards advanced adaptive grid generation techniques, such as boundary fitted curvilinear systems [1,2],

---

* Correspondence to: Department of Engineering Science, University of Oxford, Parks Road, Oxford OX1 3PJ, U.K.

Received July 1999
Revised December 1999

unstructured advancing front and Voronoi methods [3], and hierarchical grid techniques [4,5]. Such grid generators ideally should be automatic (i.e. require only boundary and flow feature information for generation), robust (i.e. work in all situations), efficient (i.e. not too expensive in computer storage and CPU time) and fit the flow boundaries correctly. Hierarchical grid generation appears an attractive option in that it meets most of the foregoing requirements. Two-dimensional quadtree (Q-tree) [6] or tritree [7] grids may be produced extremely quickly by simple spatial decomposition into rectangles or triangles about seeding point information.

Accurate solutions of the SWEs can be achieved using Godunov-type schemes, which are appropriate for systems of hyperbolic equations and are applicable to discontinuous flow situations, such as those occurring at hydraulic jumps [8–10]. Special care needs to be taken in properly discretizing the convection terms at cell interfaces, while damping non-physical oscillations by means of a limiter [10,11]. Several authors have experienced problems in solving the hyperbolic SWEs using Roe's approximate Riemann solver because of the difficulty in balancing the flux gradient and source terms. For example, Ambrosi [12] observed that a problem occurred when simulating still water above an uneven bed. Nugic [13] also noted that very poor results were obtained for cases of shallow flow with variable depth and referred to the problem as numerical incompatibility. Nugic proposed a revised mathematical formulation of the SWEs, by reallocating all bed slope related flux gradients to the source term. Recently, Vázquez-Cendón [14] used numerical upwinding of the source terms to rectify the equilibrium between flux gradient and source terms. Hubbard and Garcia-Navarro [15] have since extended this numerical treatment to higher-order total variation diminishing (TVD) schemes.

A new conservative hyperbolic version of the non-linear SWEs is derived in the present paper, properly balancing source and flux gradient terms. Hence, the need for *a posteriori* numerical treatment is removed. This paper first gives a brief description of Q-tree grid generation. The equations are written in Cartesian matrix form suitable for application on rectangular Q-tree grids. The equations are discretized using finite volumes collocated with the grid cells, and integrated in time using a second-order Adams–Bashforth scheme. Results are presented for a series of validation tests designed to verify the accuracy and efficiency of the adaptive Q-tree model. The tests include simulations of wind-induced flow in a basin with non-uniform bathymetry, moving fronts (i.e. the dam-break problems), flow recirculation (jet-forced flow in a reservoir) and a shock-type flow discontinuity (the oblique hydraulic jump).

2. Q-TREE GRID GENERATION

Q-tree grid generation is achieved by normalizing the problem domain so that it fits into a unit square, sub-dividing the unit square (root panel) into four quadrants, searching for seeding points and sub-dividing panels in which a prescribed number are found, and finally imposing grid regularization. In this case, grid regularization is achieved by ensuring a maximum 2:1 ratio between the linear dimensions of adjacent cells, and by limiting the number of cell configurations in the grid so that it has greater uniformity. The 2:1 condition provides a trade-off between grid flexibility and accuracy. For example, no more than one hanging node can occur on any cell edge at the interface between cells of different size. This reduces the possible interpolation error through discretization, while allowing the grid density to vary significantly over relatively short distances. The regularized regions spread out from areas of high cell density, thus minimizing the number of neighbour cells checked for compliance of the 2:1 condition.

Q-tree grid information is held in integer arrays with pointers to parent and child cells. Similar to the method proposed by Van Dommelen and Rundensteiner [16], each cell is given a unique identification number consisting of a concatenated series of 1s and 2s and trailing zeros according to its position in the unit square. The identification number is built-up from the root (i.e. unit square) level during the sub-division process; at any arbitrary level, the numbers 11, 12, 22, 21 refer to northwest, northeast, southeast and southwest respectively. Cells that are at a lower level of resolution than the prescribed finest level are given trailing zeros at the end of their identification numbers so that they have the same total number of integers as that of a cell at the finest level. Consider Figure 1; by concatenation of the local reference numbers and addition of trailing zeros, the identification number for cell A at level 4 out of a finest level 16 is $N_A = 122112220000000000000000000000$. Cell neighbour finding and grid regularization are jointly undertaken, making use of the Q-tree structure. As indicated by Greaves and Borthwick [17], it is straightforward to determine the identification numbers of parent, children and same-size adjacent cells through simple manipulations of the identification number. Neighbours are identified by checking for the presence of same-size adjacent (both sideways and corner) cells, as if the grid were uniform. If the cell lies within a non-uniform grid cell configuration, neighbour finding involves searching the tree for the nearest common ancestor shared between the cell of interest and adjacent/corner neighbours. Neighbours are then found by searching the tree from the nearest common ancestor to leaves at the highest division level.

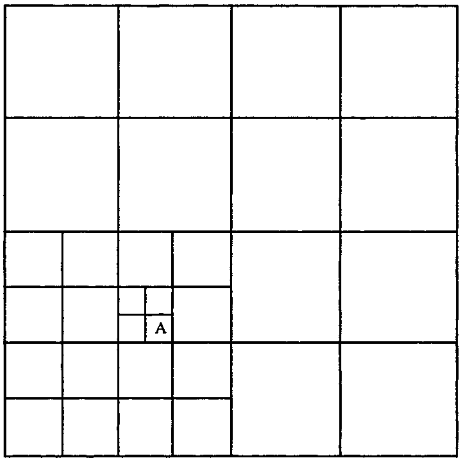

Figure 1. Typical Q-tree grid arrangement.

Grid refinement and coarsening are achieved by the selective addition and removal of Q-tree cells according to user specified criteria, including boundary seeding points formed from the discretized geometry, the free surface gradient and the magnitude of depth-averaged vorticity.

3. GOVERNING EQUATIONS

The SWEs describe flow in shallow water bodies where the vertical acceleration within the fluid is negligible and the pressure assumed hydrostatic. In differential form, the SWEs are written as

$$\frac{\partial \zeta}{\partial t} + \frac{\partial (uh)}{\partial x} + \frac{\partial (vh)}{\partial y} = 0$$

$$\frac{\partial (uh)}{\partial t} + \frac{\partial (u^2 h)}{\partial x} + \frac{\partial (uvh)}{\partial y} - v\left(\frac{\partial (hu_x)}{\partial x} + \frac{\partial (hu_y)}{\partial y}\right) = \frac{\tau_{wx} - \tau_{bx}}{\rho} - gh\frac{\partial \zeta}{\partial x} + hfv$$

$$\frac{\partial (vh)}{\partial t} + \frac{\partial (uvh)}{\partial x} + \frac{\partial (v^2 h)}{\partial y} - v\left(\frac{\partial (hv_x)}{\partial x} + \frac{\partial (hv_y)}{\partial y}\right) = \frac{\tau_{wy} - \tau_{by}}{\rho} - gh\frac{\partial \zeta}{\partial y} - hfu \tag{1}$$

where ζ is the free surface elevation above the still water level h_s, $h\ (= \zeta + h_s)$ is the total water depth, u and v are depth-averaged velocities in the x- and y-directions respectively, u_x, u_y and v_x, v_y are derivatives of the depth-averaged velocity components in the x- and y-directions respectively, g is acceleration due to gravity, ρ is water density, τ_{wx} and τ_{wy} are surface stresses, τ_{bx} and τ_{by} are bed friction stresses, f is the Coriolis parameter and v is the kinematic eddy viscosity coefficient. It is assumed also in the above equations that the vertical viscous forces are negligible, and that the momentum correction factor due to the non-uniform vertical distribution of horizontal velocities can be taken as unity. In many nearly horizontal flows, these approximations are reasonable (see e.g. Reference [18]).

The $gh\,\partial\zeta/\partial x$ term is usually split to give the hyperbolic formulation used by many researchers (e.g. References [3,19,20])

$$gh\frac{\partial \zeta}{\partial x} = \frac{1}{2}g\frac{\partial (h^2)}{\partial x} + ghS_{ox} \tag{2}$$

where S_{ox} is the bed slope in the x-direction. However, when applied to non-uniformly varying bathymetries, this approach leads to problems conserving continuity and momentum for individual cells when using Roe's approximate Riemann solver. The split of the free surface gradient term into source and flux gradient components causes an imbalance, despite the fact that the finite volume scheme is inherently conservative. This problem is independent of geometric configuration. A consequence of using the conventional formulation is that still water cannot be computed; hence, the scheme produces physically meaningless results when applied to cases with uneven beds. The difficulty is overcome herein by splitting and redistributing the $gh\,\partial\zeta/\partial x$ term in a novel way, which still retains the hyperbolic nature of the SWEs

$$gh\frac{\partial \zeta}{\partial x} = \frac{1}{2}g\frac{\partial(\zeta^2 + 2\zeta h_s)}{\partial x} + g\zeta S_{ox} \tag{3}$$

Applying a similar operation to the $gh\,\partial\zeta/\partial y$ term, the SWEs are rewritten as

$$\frac{\partial \zeta}{\partial t} + \frac{\partial(uh)}{\partial x} + \frac{\partial(vh)}{\partial y} = 0$$

$$\frac{\partial(uh)}{\partial t} + \frac{\partial(u^2h + \frac{1}{2}g(\zeta^2 + 2\zeta h_s))}{\partial x} + \frac{\partial(uvh)}{\partial y} - v\left(\frac{\partial(hu_x)}{\partial x} + \frac{\partial(hu_y)}{\partial y}\right) = \frac{\tau_{wx} - \tau_{bx}}{\rho} - g\zeta S_{ox} + hfv$$

$$\frac{\partial(vh)}{\partial t} + \frac{\partial(uvh)}{\partial x} + \frac{\partial(v^2h + \frac{1}{2}g(\zeta^2 + 2\zeta h_s))}{\partial y} - v\left(\frac{\partial(hv_x)}{\partial x} + \frac{\partial(hv_y)}{\partial y}\right) = \frac{\tau_{wy} - \tau_{by}}{\rho} - g\zeta S_{oy} - hfu \tag{4}$$

where S_{oy} is the bed slope in the y-direction. Other mathematical formulations are possible, but when constructing the system's flux Jacobian and discretizing using finite volumes, the equations can produce non-hyperbolic and unconditionally unstable results. This formulation allows Roe's approximate Riemann solver to be used to evaluate the inviscid flux between adjacent cells for all bathymetries.

4. NUMERICAL SOLVER

The two-dimensional SWEs are discretized spatially using finite volumes on the Q-tree grid, with Roe's flux function used to represent the non-linear convection terms. Time integration is by means of the second-order Adams–Bashforth technique. Unwanted oscillations are avoided using a non-linear limiter.

The SWEs can be written in integral form as

$$\frac{\partial}{\partial t}\int_{\Omega} \mathbf{q}\, \mathrm{d}\Omega + \int_{\Omega}\left(\frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{g}}{\partial y}\right)\mathrm{d}\Omega = \int_{\Omega} \mathbf{h}\, \mathrm{d}\Omega \tag{5}$$

where Ω is the problem domain, $\mathbf{q}$ is the vector of conserved variables, $\mathbf{f}$ and $\mathbf{g}$ are flux vectors, and $\mathbf{h}$ is the vector of forcing functions. The vectors $\mathbf{q}$, $\mathbf{f}$, $\mathbf{g}$ and $\mathbf{h}$ are given by

$$\mathbf{q} = \begin{bmatrix} \zeta \\ uh \\ vh \end{bmatrix}, \qquad \mathbf{f} = \begin{bmatrix} uh \\ u^2h + g(\zeta^2 + 2\zeta h_s)/2 - vh\,\partial u/\partial x \\ uvh - vh\,\partial v/\partial x \end{bmatrix}$$

$$\mathbf{g}=\begin{bmatrix} vh \\ uvh - \nu h\,\partial u/\partial y \\ v^2h + g(\zeta^2+2\zeta h_s)/2 - \nu h\,\partial v/\partial y \end{bmatrix}, \qquad \mathbf{h}=\begin{bmatrix} 0 \\ (\tau_{wx}-\tau_{bx})/\rho - g\zeta S_{ox} + hfv \\ (\tau_{wy}-\tau_{by})/\rho - g\zeta S_{oy} - hfu \end{bmatrix}$$

By applying Green's theorem to the second term in Equation (5)

$$\frac{\partial}{\partial t}\int_\Omega \mathbf{q}\,\mathrm{d}\Omega + \oint_S \hat{\mathbf{f}}\,\mathrm{d}S = \int_\Omega \mathbf{h}\,\mathrm{d}\Omega \tag{6}$$

where S is the boundary of Ω, and $\hat{\mathbf{f}}$ is the vector of flux functions through S given by

$$\hat{\mathbf{f}} = \mathbf{f}n_x + \mathbf{g}n_y \tag{7}$$

in which n_x and n_y are the Cartesian components of $\mathbf{n}$, the unit normal vector to S. Equation (7) may also be written in terms of inviscid and viscous fluxes as

$$\hat{\mathbf{f}} = \mathbf{f}^{\mathrm{I}} - \nu\mathbf{f}^{\mathrm{V}} \tag{8}$$

where

$$\mathbf{f}^{\mathrm{I}}=\begin{bmatrix} uhn_x + vhn_y \\ (u^2h + g[\zeta^2+2\zeta h_s]/2)n_x + uvhn_y \\ uvhn_x + (v^2h + g[\zeta^2+2\zeta h_s]/2)n_y \end{bmatrix}$$

and

$$\mathbf{f}^{\mathrm{V}}=\begin{bmatrix} 0 \\ (h\,\partial u/\partial x)n_x + (h\,\partial u/\partial y)n_y \\ (h\,\partial v/\partial x)n_x + (h\,\partial v/\partial y)n_y \end{bmatrix}$$

The equations are discretized on a collocated grid with uh, vh and ζ stored at the centre of each cell. The location of variables in an arbitrary control volume is shown in Figure 2. For each cell, Equation (6) can be more conveniently rewritten as

$$\left.\frac{\partial V\mathbf{q}}{\partial t}\right|_i = -\oint_{\partial C_i} \hat{\mathbf{f}}_i\,\mathrm{d}s + V_i\mathbf{h}_i \tag{9}$$

where $\mathbf{q}_i$ and $\mathbf{h}_i$ are the cell centre values, and V_i denotes the area of cell i. For the Q-tree cells, ∂C_i is the path of integration, which is along all four cell edges where the flux out of each face is assumed uniform. The surface integral in Equation (9) can be evaluated in discrete form by using the equation

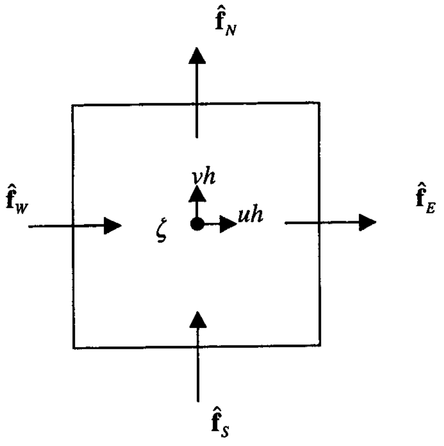

Figure 2. Control volume showing location of variables.

$$\oint_{\partial C_i} \hat{\mathbf{f}}_i \,\mathrm{d}s = (\hat{\mathbf{f}}_\mathrm{E} - \hat{\mathbf{f}}_\mathrm{W} + \hat{\mathbf{f}}_\mathrm{N} - \hat{\mathbf{f}}_\mathrm{S})\Delta s \tag{10}$$

where $\hat{\mathbf{f}}_\mathrm{E}$, $\hat{\mathbf{f}}_\mathrm{W}$, $\hat{\mathbf{f}}_\mathrm{N}$ and $\hat{\mathbf{f}}_\mathrm{S}$ are the vector fluxes through the east, west, south and north faces of each cell respectively, and Δs is the length of the cell side.

4.1. *Inviscid fluxes*

The inviscid fluxes can be evaluated by adopting Roe's approximate Riemann solver at each cell edge as follows:

$$\mathbf{f}^\mathrm{I}_{i,j} = \frac{1}{2}\left[\mathbf{f}^\mathrm{I}(\mathbf{q}^+_{i,j}) + \mathbf{f}^\mathrm{I}(\mathbf{q}^-_{i,j}) - |\mathbf{A}|(\mathbf{q}^+_{i,j} - \mathbf{q}^-_{i,j})\right] \tag{11a}$$

in which

$$|\mathbf{A}| = \mathbf{R}|\Lambda|\mathbf{L} \tag{11b}$$

where $\mathbf{q}^+_{i,j}$ and $\mathbf{q}^-_{i,j}$ are reconstructed right and left Riemann states respectively at the cell interface, located between adjacent cells i and j, $\mathbf{A}$ is the flux Jacobian evaluated using $\mathbf{R}$ and $\mathbf{L}$, the right and left eigenvector matrices of $\mathbf{A}$ respectively, and $|\Lambda|$ is a diagonal matrix of the absolute values of the eigenvalues of $\mathbf{A}$. The inviscid flux Jacobian is given by

$$\mathbf{A} = \frac{\partial(\mathbf{f}^\mathrm{I})}{\partial \mathbf{q}} = \begin{bmatrix} 0 & n_x & n_y \\ (c^2 - u^2)n_x - uvn_y & 2un_x + vn_y & un_y \\ -uvn_x + (c^2 - v^2)n_y & vn_x & un_x + 2vn_y \end{bmatrix} \tag{12}$$

which has real and distinct eigenvalues (confirming hyperbolicity) given by

$$\begin{aligned} \lambda_1 &= un_x + vn_y \\ \lambda_2 &= un_x + vn_y - cn \\ \lambda_3 &= un_x + vn_y + cn \end{aligned} \tag{13}$$

where c is the wave celerity and $n = \sqrt{n_x^2 + n_y^2}$. The associated right eigenvector matrix is

$$\mathbf{R} = \begin{bmatrix} 0 & 1 & 1 \\ n_y & u - \left(\dfrac{cn_x}{n}\right) & u + \left(\dfrac{cn_x}{n}\right) \\ -n_x & v - \left(\dfrac{cn_y}{n}\right) & v + \left(\dfrac{cn_y}{n}\right) \end{bmatrix} \tag{14}$$

and the left eigenvector matrix is given by

$$\mathbf{L} = \begin{bmatrix} -\dfrac{(un_y - vn_x)}{n^2} & \dfrac{n_y}{n^2} & \dfrac{n_x}{n^2} \\ \dfrac{1}{2} + \dfrac{un_x + vn_y}{2cn} & \dfrac{-n_x}{2cn} & \dfrac{-n_y}{2cn} \\ \dfrac{1}{2} - \dfrac{un_x + vn_y}{2cn} & \dfrac{n_x}{2cn} & \dfrac{n_y}{2cn} \end{bmatrix} \tag{15}$$

Despite the different formulation given in Equation (4), the flux Jacobian **A**, its eigenvectors and associated right and left eigenvector matrices, are identical to those obtained using the conventional approach (see, e.g. Reference [3]). The variables u, v, c in Equations (12)–(15) are given by Roe's average state, which is defined as

$$u = \frac{u^+\sqrt{h^+} + u^-\sqrt{h^-}}{\sqrt{h^+} + \sqrt{h^-}}, \qquad v = \frac{v^+\sqrt{h^+} + v^-\sqrt{h^-}}{\sqrt{h^+} + \sqrt{h^-}} \quad \text{and}$$

$$c = \sqrt{\frac{g(h^+ + h^-)}{2}} \tag{16}$$

where the superscripts $+$ and $-$ denote the right and left Riemann states on either side of a cell interface respectively.

As h increases, so does $c = \sqrt{gh}$, and hence $\lambda_1 = a$ (where $a = un_x + vn_y$), $\lambda_2 \to -c$ and $\lambda_3 \to c$. By analogy with a system of ordinary differential equations (ODEs), the stiffness ratio (Toro [10], p. 504) tends towards c/a. In other words, the ratio between the shallow water wave celerity and the depth-averaged flow speed is an approximate guide to the system's stiffness. Toro notes that serious numerical difficulties may arise in explicit methods as the stiffness increases above 20. For cases starting from rest, this is an obviously important consideration. However, problems were not encountered using the present scheme within the range of test cases. It should be noted that the shallow water approximation $h/L \ll 1$, where L is the free surface wavelength, becomes violated at high values of h (and hence c).

4.2. Limiter

In order to evaluate Equations (12)–(15), the Riemann states either side of the interface must be determined. Spatial information on the velocities and depth within the cell can be obtained from an interpolation across an arbitrary cell A given by

$$q(x, y) = q_A + (\nabla q_A) \cdot \mathbf{r} \tag{17}$$

where q is a component of the vector of conserved variables $\mathbf{q}$, and q_A is its value at the centre of cell A, ∇ is the gradient vector and $\mathbf{r}(x, y)$ is the vector from the cell centre to any arbitrary point within the cell. Equation (17) is a piecewise linear interpolation that gives a second-order spatial approximation.

However, numerical oscillations can occur when evaluating ∇q_A. It is highly desirable to avoid creating numerical oscillations when trying to simulate flow discontinuities. The main options available for controlling or limiting oscillations around discontinuities are to use artificial viscosity or non-linear limiters. Slope limiters suppress the generation of local extrema in a non-linear manner when the variables u, v, h are reconstructed for use in Equations (12)–(15). Herein, a slope limiter is implemented such that for consecutive cells $i-1$, i, $i+1$ on a locally uniform grid (using interpolation if necessary), Equation (17) is replaced by

$$q(x, y) = q_i + \frac{\Phi}{2}(q_i - q_{i-1}) \tag{18}$$

where Φ is the slope limiter. In practice, the limiter restricts the gradient such that the cell-face data, reconstructed from the stored cell-centre data, falls within the range of the cell-centre data in the cells either side of the interface. As given by Hirsch [11], Φ is defined as

$$\Phi(r) = \max[0, \min(\beta r, 1), \min(r, \beta)] \tag{19}$$

where the limiter parameter, β, is restricted to $1 \leq \beta \leq 2$, and the gradient ratio is given by

$$r = \frac{q_{i+1} - q_i}{q_i - q_{i-1}} \tag{20}$$

Equations (18)–(20) are defined such that if $r > 1$, a slope of $(q_i - q_{i-1})/|\mathbf{r}|$ is used to predict the Riemann state, or if $r < 1$ the selected slope is $(q_{i+1} - q_i)/|\mathbf{r}|$.

In particular, $\beta = 1$ gives the minmod limiter, and $\beta = 2$ gives Roe's Superbee limiter. The choice of minmod slope limiter used herein is of relatively low strength and so is slightly dissipative, but this effect should not be overly significant for shallow water flows. It should be noted that Anastasiou and Chan [3] have compared the effect of using different slope limiters, and found that as the limiter increases in strength it is less diffusive but permits greater numerical oscillations.

4.3. Boundary conditions

The discretization of the SWEs applies to cells in the interior of the computational mesh. For cells on the boundary of the domain, special discretizations of the SWEs have to be applied to stabilize the numerical scheme and maintain continuity.

Herein, slip or no-slip conditions may be applied at closed boundaries. In both cases the velocity normal to the wall is set to zero to represent no flux through the boundary. To describe the no-slip boundary, the tangential velocity component at the cell edge is also set equal to zero. Riemann invariants, specified according to the local Froude number, have been used to implement open boundary conditions as follows:

(i) sub-critical flow (when the water depth h_B is imposed)

$$\tilde{u}_B = \tilde{u}_\mathrm{I} + 2\sqrt{g}(\sqrt{h_\mathrm{I}} - \sqrt{h_B}), \qquad \tilde{v}_B = \tilde{v}_\mathrm{I} \tag{21a}$$

(ii) sub-critical flow (when the velocity $\tilde{u}_B$ is imposed)

$$h_B = \left(\frac{1}{2\sqrt{g}}(\tilde{u}_\mathrm{I} - \tilde{u}_B) + \sqrt{h_\mathrm{I}}\right)^2, \qquad \tilde{v}_B = \tilde{v}_\mathrm{I} \tag{21b}$$

(iii) supercritical inflow (the variables $(h_B, \tilde{u}_B, \tilde{v}_B)$ are given); and
(iv) supercritical outflow

$$h_B = h_\mathrm{I}, \qquad \tilde{u}_B = \tilde{u}_\mathrm{I}, \qquad \tilde{v}_B = \tilde{v}_\mathrm{I} \tag{21c}$$

In Equation (21), subscript B denotes the boundary value, subscript I denotes the inner Riemann state value at the boundary, $\tilde{u}$ and $\tilde{v}$ are the depth-averaged velocity components normal and tangential to the boundary respectively.

4.4. Time integration

A second-order Adams–Bashforth time stepping numerical solver is employed to integrate the SWEs in time. At time $t = n\Delta t$, where Δt is the time step, Equation (9) may be written as

$$\left.\frac{\partial V\mathbf{q}}{\partial t}\right|_i^n = -\oint_{\partial C_i} \hat{\mathbf{f}}_i^n \, \mathrm{d}s + V_i^n \mathbf{h}_i^n \tag{22}$$

in which the superscript n refers to the time level. Then, the Adams–Bashforth method is

$$\mathbf{q}_i^{n+1} = \mathbf{q}_i^n + \frac{\Delta t}{V_i^n}\left(\frac{3}{2}\left.\frac{\partial V\mathbf{q}}{\partial t}\right|_i^n - \frac{1}{2}\left.\frac{\partial V\mathbf{q}}{\partial t}\right|_i^{n-1}\right) \tag{23}$$

The time step was selected by trial and error. This is because the allowable time step for stability is not only affected by a Courant criterion based on cell size and local flow speed (often the smallest cell size being critical). It is also affected by stability of the Adams–Bashforth method and the time taken for numerical waves resulting from the evolution of discontinuities from independent Riemann problems to interact. The time step was fixed *a priori*, and then left unchanged throughout computations. It was chosen so that it did not violate any of the foregoing stability constraints, and so a possible disadvantage of fixed time step adaptive Q-tree grid methods is that the allowable time step may become very small. Furthermore, the Q-tree scheme presented here was not utilized at optimum efficiency, whereby different time steps could be used at different grid resolution levels.

4.5. Interpolation

The generation of Q-tree grids can produce adjacent panels of different sizes so that a hanging node is created at the interface between two cells. This requires values to be known at grid locations where the information is not stored. Thus, complicated interpolation procedures are required to approximate the values for all the different grid configurations encountered. Various techniques have been proposed to overcome this inherent disadvantage of Q-tree grids (e.g. References [21,22]). From such approaches, it is clear that a compromise is needed between sufficient accuracy of interpolation and the complexity of the technique implemented. In this paper, linear interpolation has been used as it provides sufficient accuracy, and is efficient to implement.

5. RESULTS

5.1. Wind-induced circulation in a circular basin

This test problem consists of a uniform wind stress applied to the surface of shallow water in a circular basin. The basin has a non-uniform bed profile, with still water depths that become progressively deeper towards its centre. Figure 3 shows the bathymetry of the basin with the still water depth contours given by

$$h_s = \frac{1}{1.3}\left(\frac{1}{2} + \sqrt{\left(\frac{1}{2} - \frac{1}{2}\frac{r_b}{R_0}\right)}\right) \tag{24}$$

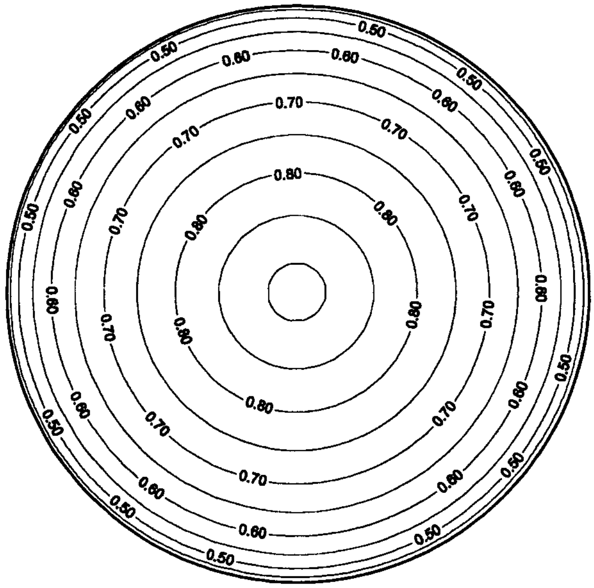

Figure 3. Circular basin: bathymetry (still water depth contours in m).

where r_b is the distance from the centre of the basin and R_0 is the radius of the basin which is equal to 193.2 m (in the case considered here). In shallow water with non-uniformly varying bathymetry, the depth-averaged velocities in deeper water can be directed against the direction of the wind, whereas in shallower water, the velocities are more likely to be in the same direction as the applied wind stress. Kranenburg [23] developed an analytical solution for this problem assuming the centre of the gyres lay on the centreline of the basin and predicted flow against the wind direction in the centre of the basin.

Initially, the water is at rest. A uniform wind stress from southwest to northeast is then applied, being increased from 0 to 0.02 $\mathrm{N\,m^{-2}}$ over a period of 1000 s to avoid numerical seiching, according to

$$\tau_w = \left(-2\left(\frac{t}{T_r}\right)^3 + 3\left(\frac{t}{T_r}\right)^2 \right)\tau_{\mathrm{ss}}, \quad t \le T_r \tag{25a}$$

$$\tau_w = \tau_{\mathrm{ss}}, \quad t \ge T_r \tag{25b}$$

where t is time, T_r is the length of time required to build up the flow and τ_{ss} is the steady state value of the wind stress. In this case, T_r is set to 1000 s and τ_{ss} is set to 0.02 $\mathrm{N\,m^{-2}}$. The bottom friction and Coriolis parameter are set to 0, and slip conditions are applied at the boundary. A constant depth-averaged eddy viscosity of 0.00012 $\mathrm{m^2\,s^{-1}}$ is used throughout the basin. A time step $\Delta t = 0.3$ s is used to ensure stability, which is below the value implied by a Courant condition. The test was run for 10000 s. To check for grid convergence, results were obtained on uniform level 6 and level 7 Q-tree grids. It was observed that the depth-averaged velocity vectors were very similar, except immediately adjacent to the boundaries where the effects of the staircase approximation were evident. Figure 4 depicts the steady state depth-averaged velocity vectors in the basin obtained using the level 7 grid, but with one in three vectors plotted for ease of visual representation. Two counter-rotating gyres can be discerned with the flow across the deepest part of the basin directed against the direction of the wind. The centres of the gyres lie slightly downwind of the north–south centreline of the basin, in agreement with the results of Cruz [22].

Figure 5 shows a typical non-uniform Q-tree grid used for the present computation, where the smallest cells are of sub-division level 8 corresponding to $\Delta x = 2.5$ m. The centre of the domain has cells with a minimum level of 6. Figures 6 and 7 show the depth-averaged velocity vectors and streamfunction contours at steady state, which confirm the position of the gyres within the basin. The flow pattern is diametrically symmetric with respect to the wind direction. The centres of the gyres lie in almost exactly the same locations as for the uniform level 7 grid discussed previously. The streamlines resemble Kranenburg's [23] results. Further comparison between the models was achieved by plotting the normalized depth-averaged velocity

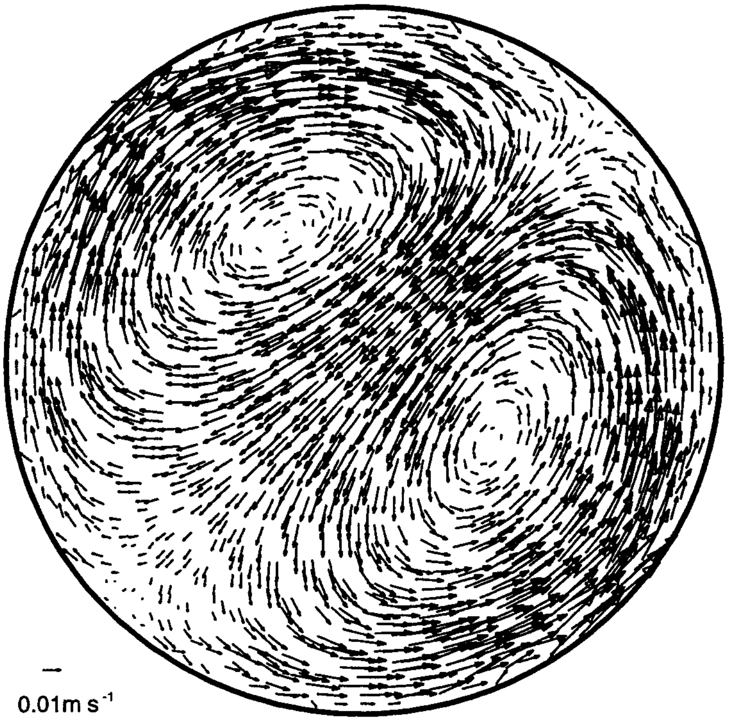

Figure 4. Circular basin: steady state wind-induced velocities on uniform level 7 Q-tree grid.

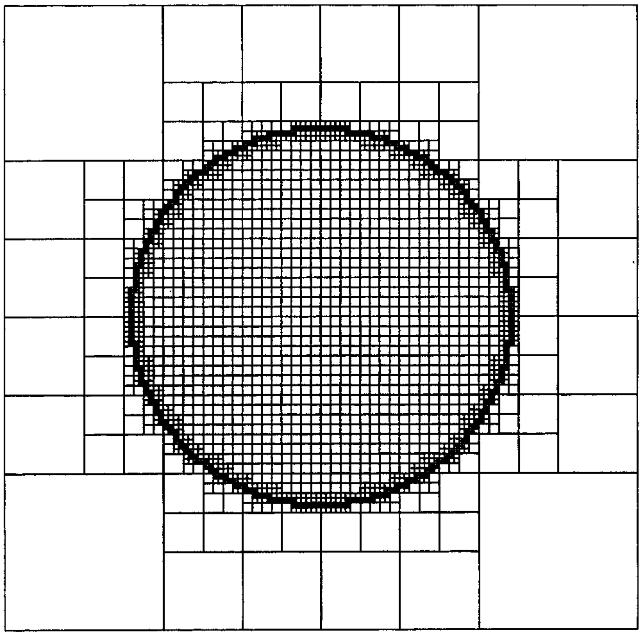

Figure 5. Circular basin: non-uniform Q-tree grid.

$$\frac{(u+v)}{\sqrt{2}}\frac{\kappa}{u_* \ln Z}$$

perpendicular to the wind direction across the centre of the basin. Note that $\kappa = 0.4$ is the von Kármán constant, u_* is the shear velocity at the free surface ($u_* = \sqrt{\tau_w/\rho}$, where $\tau_w = \sqrt{\tau_{wx}^2 + \tau_{wy}^2}$) and $Z = H/z_0$, where the weighted mean water depth H = 1/1.3 m and z_0 is bed roughness height. Here, $z_0 = 2.8$ mm, corresponding to laminar flow over a hydraulically smooth bed, with the viscosity as given above. A comparison between the simulated profile and Kranenburg's theoretical prediction is given in Figure 8. There is reasonable agreement in the general behaviour of the normalized depth-averaged velocity, although the numerical results do not exhibit a kink at the centre of the basin and are otherwise more oscillatory than the analytical profile. The former discrepancy arises from the smoothing effect of discretizing the bed elevations near the middle of the basin. The latter oscillation may be due to the present solution attempting to rectify the central discrepancy. Further reasons why there may be slight differences between the results are the approximation of a rigid lid and assumed gyre locations in Kranenburg's model. It should also be noted that care is needed to model the curved boundaries, since in coarse grids significant amounts of spurious vorticity may be shed from the staircase representation. The root mean-square error (r.m.s.e.) was estimated for this case by examining the difference between the simulated and analytical velocity values along the centreline. For the present scheme, which is second-order accurate in space, the r.m.s.e. was

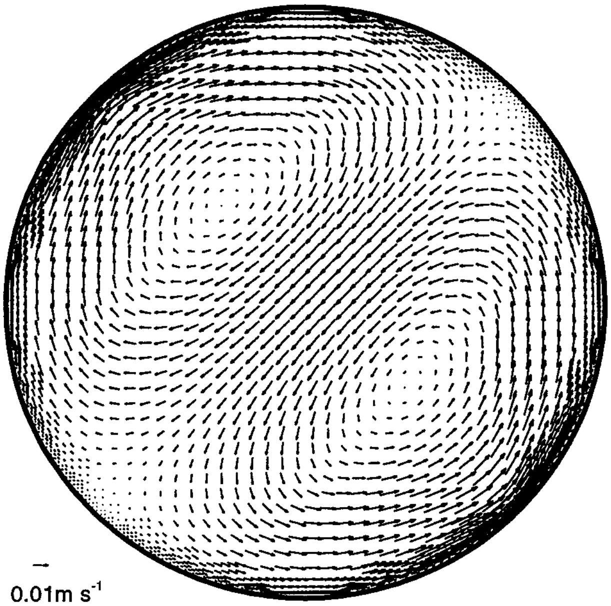

Figure 6. Circular basin: steady state wind-induced velocities on non-uniform Q-tree grid.

0.032. By setting the limiter Φ equal to zero, the scheme was rendered first-order accurate and the r.m.s.e. was found to increase to 0.134. Taken overall, this test validates the use of the new formulation given in Equation (4) in solving the SWEs using Roe's approximate Riemann solver on non-uniform bathymetries.

The computations were undertaken on a Digital Pentium 200 MHz Computer with 128 Mb RAM. Taking the results of the non-uniform Q-tree grid as an example case, it took 6 h CPU time and utilized 200000 variables in double precision for the simulation to reach $t = 10000$ s, corresponding to the steady state results depicted in Figures 6–8.

5.2. Rectangular dam-break

A square box of plan dimensions 200×200 m with a horizontal bed is divided by an infinitesimally thin impermeable wall into two equal compartments each measuring 200×100 m. The initial still water depths are 10 and 5 m either side of the dividing wall. At time $t = 0$, the dividing wall is instantaneously opened a distance of 75 m, as depicted in Figure 9. A bore then moves downstream as water discharges from the higher to the lower level. Meanwhile, a depression wave moves in the opposite direction. The dam-break problem provides a very difficult test for numerical schemes due to the discontinuous initial condition, yet is widely reported in the literature [3,19,20,24].

The initial Q-tree grid was created by normalizing the 200×200 m square box, so that it fitted exactly into the unit (root) square, and then sub-dividing uniformly to level 5 to form a

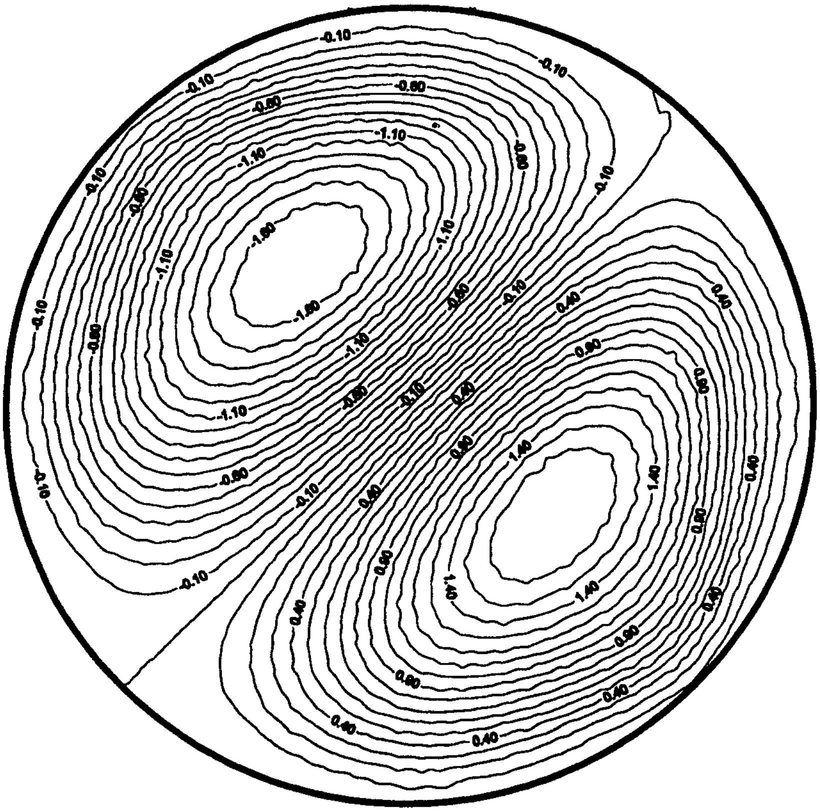

Figure 7. Circular basin: depth-averaged streamlines on non-uniform Q-tree grid.

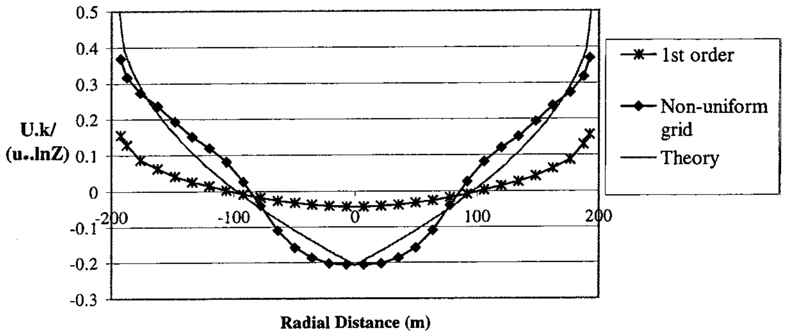

Figure 8. Circular basin: normalized depth-averaged velocity profiles.

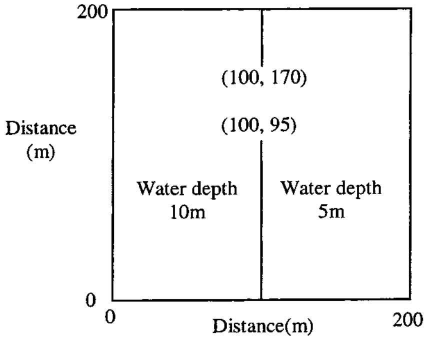

Figure 9. Rectangular dam-break: definition sketch.

6.25 m spaced square grid. The time step Δt was 0.1 s. The bed friction coefficient, Coriolis parameter and eddy viscosity coefficient were all set to zero throughout the computation.

During the time stepping procedure, the Q-tree grid adapted according to criteria based on the magnitude of the free surface gradient

$$\Theta = \sqrt{\left(\frac{\partial \zeta}{\partial x}\right)^2 + \left(\frac{\partial \zeta}{\partial y}\right)^2} \tag{26}$$

Grid enrichment is undertaken whereby cells of resolution level less than 7 are sub-divided if $\Theta > 0.1$. Grid coarsening takes place by removal of all four cells, within a given parent cell, if their resolution level is greater than 5 and the combined value of $\Theta < 0.02$. It should be noted that resolution levels 7 and 5 are user-defined notional maximum and minimum sub-division levels, respectively.

Figure 10(a)–(c) illustrates the adaptive Q-tree grids after 2, 4 and 7.2 s for mesh enrichment and coarsening. The corresponding water depth contours are presented in Figure 11. The grid evolution approximately matches the water depths indicating that the adaptation criteria are working properly. The final set of results is in almost exact agreement with other solutions presented in the literature (e.g. Reference [3]). Two vortices have formed immediately downstream and to the sides of the breach opening.

Figure 12 shows the ratio of cell numbers between uniform grid and adaptive grids (with and without grid coarsening). It is clear that adaptive Q-tree grids lead to significant efficiency gains with regard to array storage.

5.3. Circular dam-break

Another popular validation test examines the evolution of a cylindrical dam-break [3,19,25,26]. Initially, water is retained at a depth of 10 m in a cylinder of radius 11 m. The cylinder is infinitesimally thin walled, and located at the centre of a square flat-bottomed tank of side 100 m. Again, the initial condition is discontinuous in that the cylindrical wall is removed instantaneously at $t = 0$. The time step is 0.01 s.

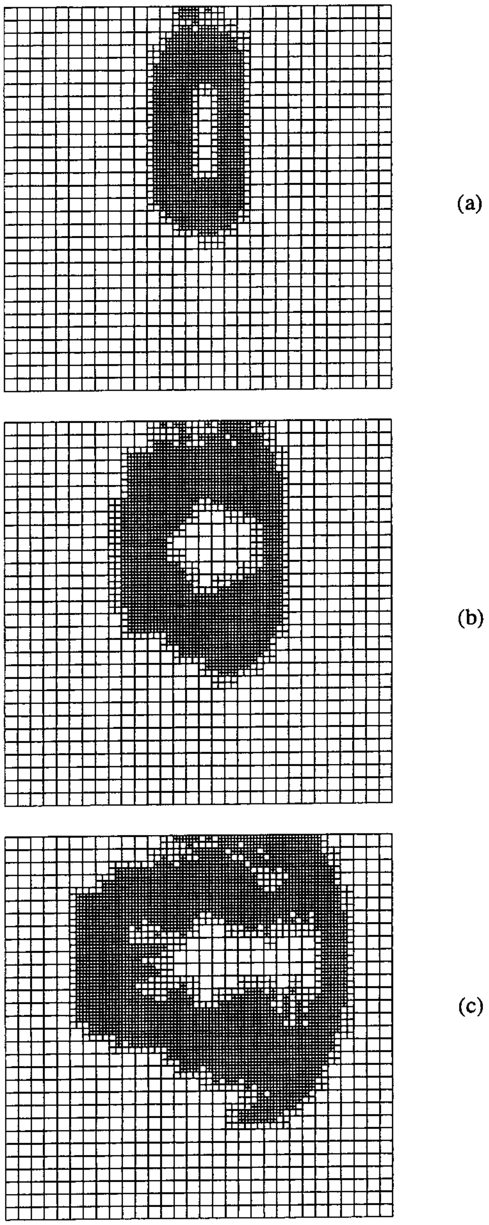

Figure 10. Rectangular dam-break: adapted Q-tree grids at $t = 2$ s (a), 4 s (b) and 7.2 s (c).

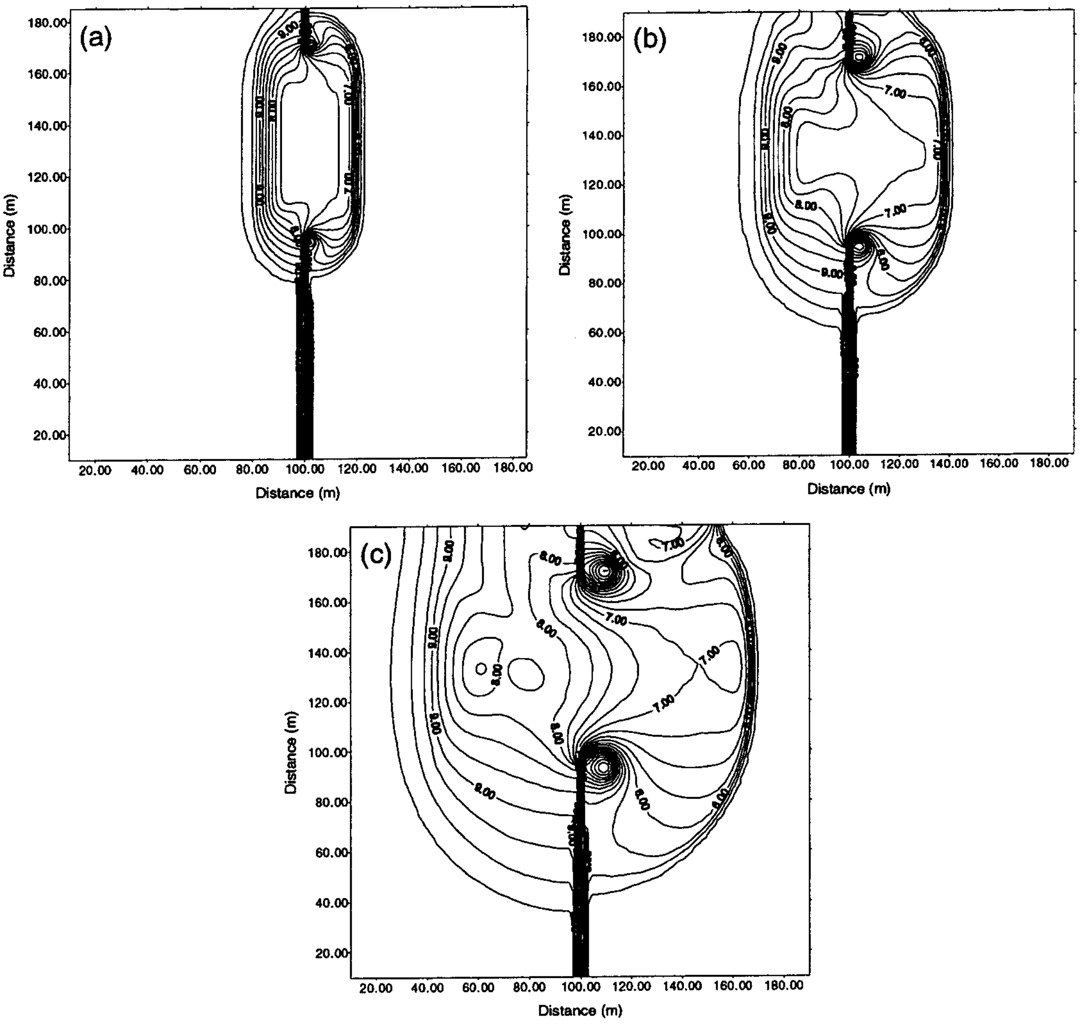

Figure 11. Rectangular dam-break: water depth contours at $t = 2$ s (a), 4 s (b) and 7.2 s (c).

Figure 13 illustrates the initial grid configuration, whereby the finest grid panels are set to level 10 with sides of 0.09765625 m. To improve computational efficiency, the maximum grid resolution level is reset to 9 at $t = 0.03$ s, 8 at $t = 0.06$ s and 7 at $t = 0.09$ s after dam-break. For grid adaptation, the magnitude of free surface gradient (defined above) is again utilized. For cells where $\Theta > 0.5$, grid enrichment is undertaken by further sub-division provided the resolution level is less than 7. Removal of cells to coarsen the grid is implemented when four cells within a parent cell have combined value of $\Theta < 0.4$, and their resolution level is greater than 5.

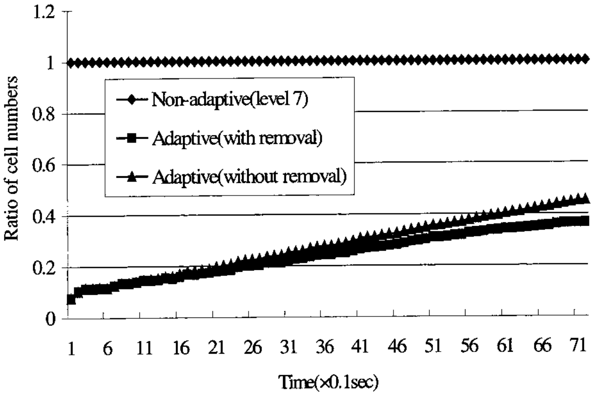

Figure 12. Rectangular dam-break: Q-tree grid cell number ratio against time.

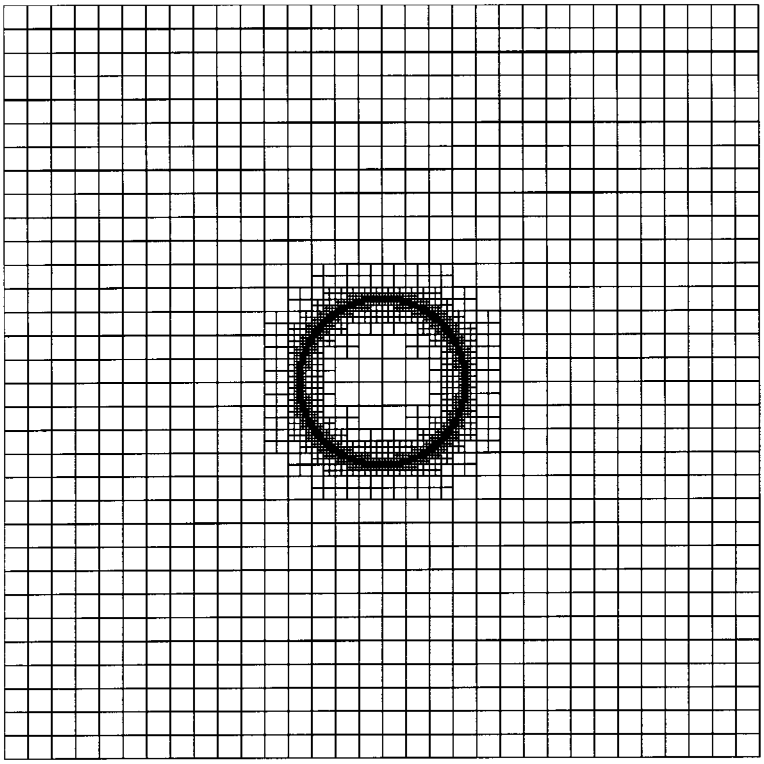

Figure 13. Circular dam-break: initial Q-tree grid.

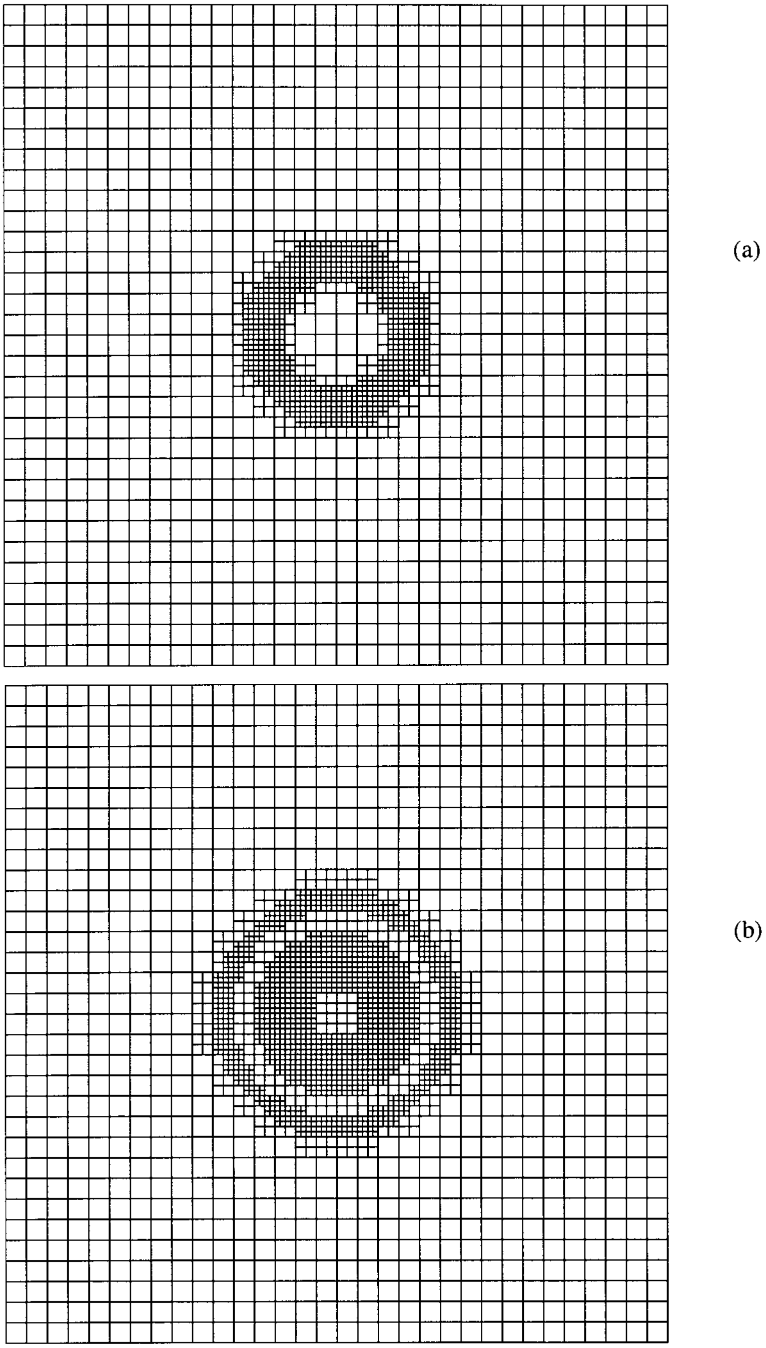

Figure 14. Circular dam-break: adapted Q-tree grids at $t = 0.2$ s (a) and 0.69 s (b).

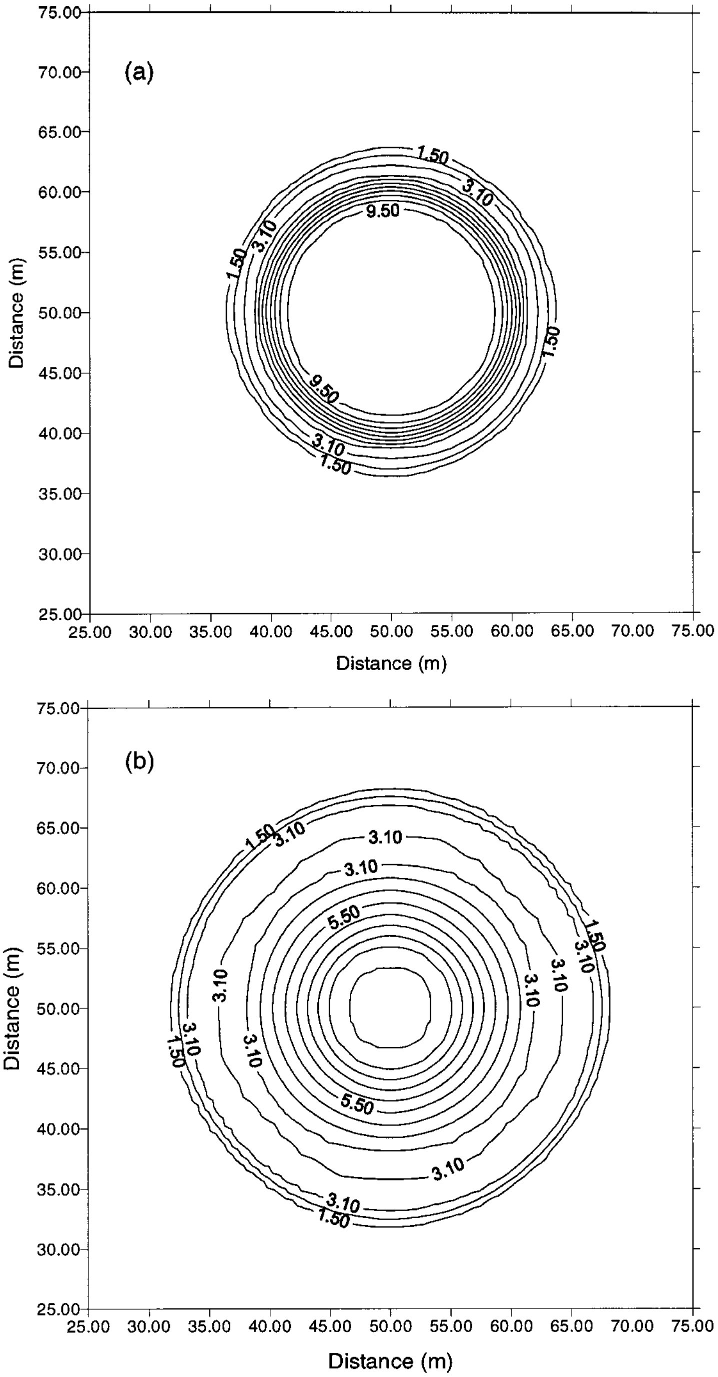

Figure 15. Circular dam-break: water depth contours at $t = 0.2$ s (a) and 0.69 s (b).

Figures 14 and 15 show the adaptive Q-tree grids and associated plan water depth contours at times $t = 0.2$ and 0.69 s. The close relationship between the adaptive grid and evolving free surface can be observed. The final set of water level contours is in excellent agreement with the predictions made by Mingham and Causon [26]. As can also be seen in the radial profile in Figure 16, the water depth falls from a central peak of 9.5 m to a circular plateau of approximately 3.0 m, which ends at the moving front of the bore.

Figure 17 presents the total number of grid cells in the adaptive Q-tree grid (with and without coarsening) as the flow evolves. Note that a uniform grid at level 7 would contain 16384 cells. The beginning stages of both adaptive Q-tree grids are similar in that they drop from a total of 4195 cells initially to 1660 cells at 0.09 s. The total number of grid cells for combined grid enrichment and coarsening is the value of 2464 cells by 0.69 s. For grid enrichment only, the numbers of grid cells continue to increase slightly, reaching 2788 cells by 0.69 s. It is again evident that adaptation has led to substantial savings in array size.

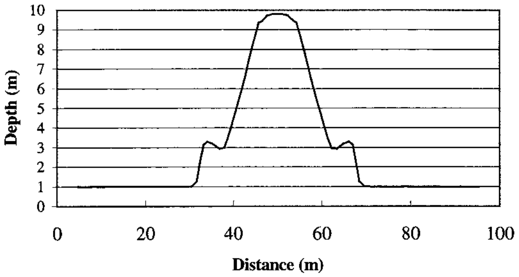

Figure 16. Circular dam-break: depth profile along a radial.

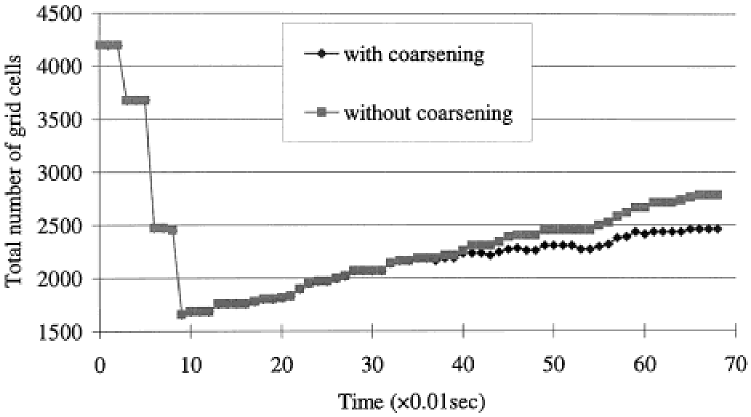

Figure 17. Circular dam-break: number of Q-tree grid cells against time.

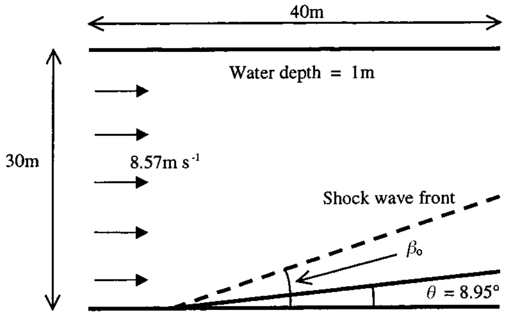

Figure 18. Oblique hydraulic jump: definition sketch.

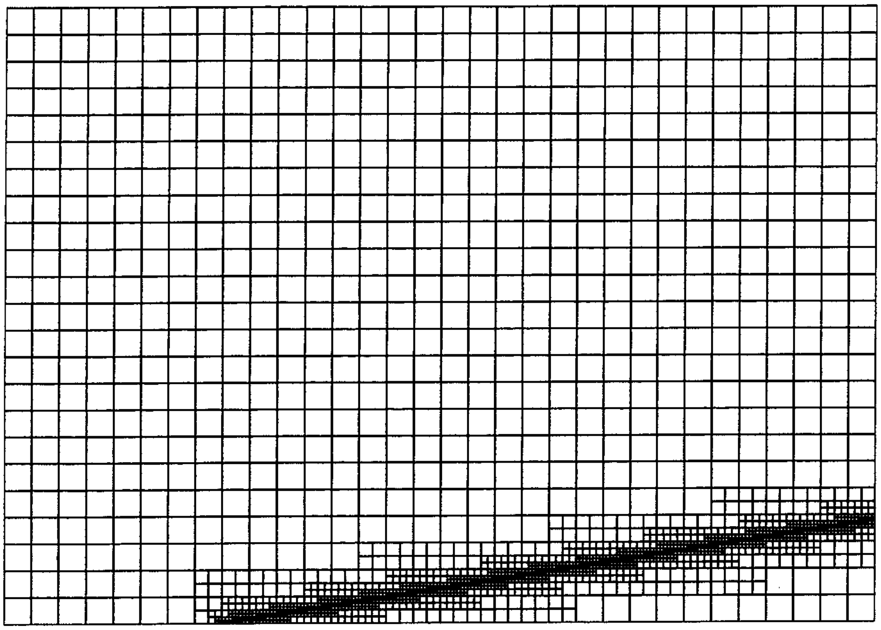

Figure 19. Oblique hydraulic jump: initial Q-tree grid.

5.4. Oblique hydraulic jump

When a supercritical flow is deflected by a converging wall (at an angle θ), the resulting shockwave forms an oblique hydraulic jump at an angle β_0, as depicted in Figure 18. This test case was considered by Anastasiou and Chan [3], Alcrudo and Garcia-Navarro [19], and Zhao *et al.* [20]. Here, the oblique hydraulic jump is computed for a channel where $\theta = 8.95°$. The

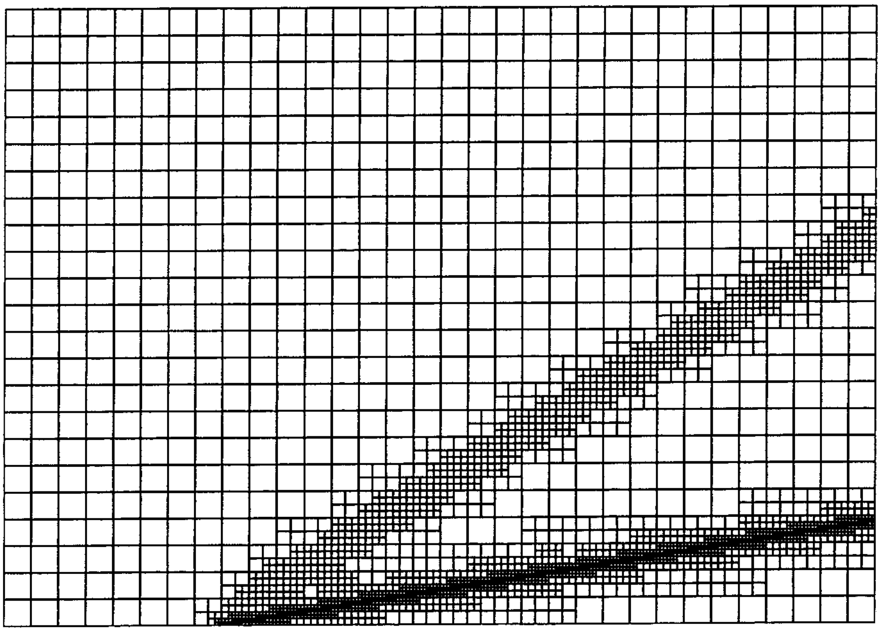

Figure 20. Oblique hydraulic jump: final adapted Q-tree grid.

initial Q-tree grid shown in Figure 19 contains 2049 cells. At the inflow boundary, the upstream water depth and the inflow velocity are 1 m and 8.57 $\mathrm{m\,s^{-1}}$ respectively. Since the resulting inflow Froude number is 2.737, the supercritical inflow condition is applied as a boundary condition. The time step was set as $\Delta t = 0.01$ s. For grid adaptation, the magnitude of free surface gradient (defined above) is again utilized. For cells where $\Theta > 0.1$, grid enrichment is undertaken by further sub-division provided the resolution level is less than 7. Removal of cells to coarsen the grid is implemented when four cells within a parent cell have a combined value of $\Theta < 0.08$, and their resolution level is greater than 5. Figure 20 shows the final 2745-cell adapted Q-tree grid obtained using the foregoing criterion. The normalized difference in water depth from one time level to the next was used to check convergence to steady state. In fact, convergence was neither very fast nor perfectly steady. By 6000 time steps (i.e. $t = 60$ s), the normalized difference remained slightly oscillatory, with an amplitude equal to 0.01 approximately.

Figures 21 and 22 give contours of water depth computed without and with grid adaptation respectively. The contour plots demonstrate that the hydraulic jump is sharply captured by the adaptive grid, and has an angle $\beta_0 \sim 30°$. In both cases, the average water depth behind the shock front is 1.53 m and the corresponding velocity is 7.90 $\mathrm{m\,s^{-1}}$. Hence, the resultant downstream Froude number is 2.04. The exact analytical solution (using the theory by Hager *et al.* [27]) is downstream water depth = 1.49984 m, downstream velocity = 7.95308 $\mathrm{m\,s^{-1}}$, downstream Froude number = 2.0737 and angle $\beta_0 = 30°$. Figure 23 depicts the free surface as

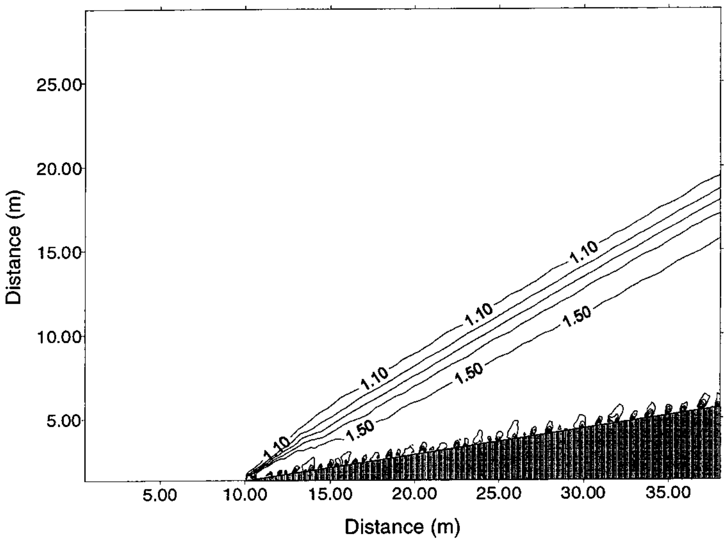

Figure 21. Oblique hydraulic jump: water depth contours (non-adapted grid).

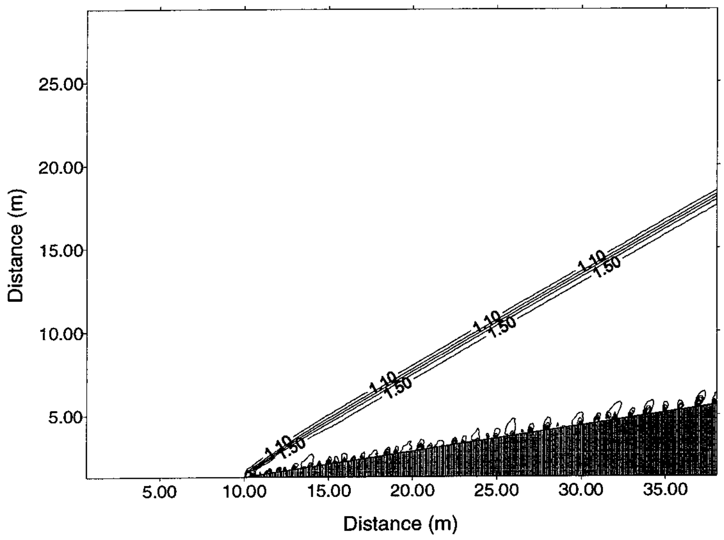

Figure 22. Oblique hydraulic jump: water depth contours (adapted grid).

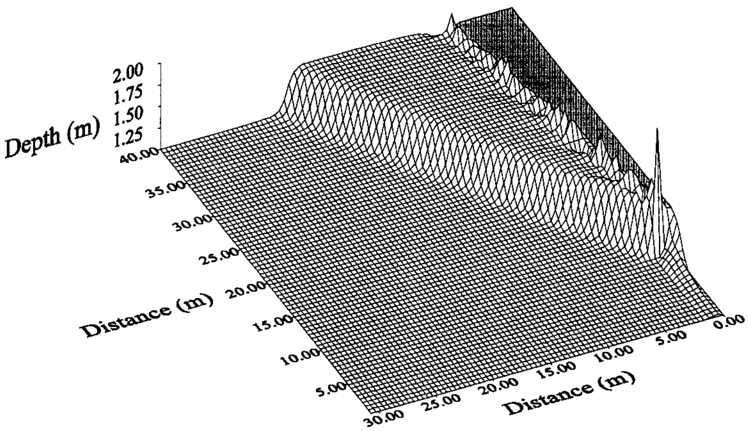

Figure 23. Oblique hydraulic jump: three-dimensional view of water surface at steady state.

a three-dimensional plot. It is obvious that the oblique jump is well defined away from the wall, but that the steady state water surface contains spikes at the convergent wall due to the lack of boundary fitting of the Q-tree grid. This boundary effect is also noticeable in the contour plots.

5.5. Jet-forced flow in a circular basin

Jet-forced flow in a shallow circular basin with a flat bed is essentially equivalent to two-dimensional viscous flow within a circle (a benchmark case that has been investigated semi-analytically by Dennis [28] and Mills [29], and also numerically, e.g. [3,30]). This test problem is useful because the geometry is complicated and unsuitable for either Cartesian or polar grids because the flow domain is bounded by straight-walled inlet and outlet stems connected to the curved-walled basin. The inlet into the reservoir is sharp-edged, and separation occurs giving rise to recirculation zones either side of a through-flow stream; the detailed flow pattern being dependent on the inlet Reynolds number. Figure 24 illustrates the initial Q-tree grid with 1940 cells. Here, the radius of the circular basin is $R_0 = 0.75$ m, and the centrelines of the inlet and outlet stems are located exactly π radians apart around the basin circumference. The openings into, and out of, the basin each subtend $2\varepsilon = \pi/15$ rad. At the beginning of the inflow stem, the inlet velocity, U_{I}, is set to 0.1 m s^{-1}. At the outlet of the outflow stem, the water depth is set at 0.1 m. The bed friction coefficient is set to zero. The inlet Reynolds number is defined as

$$Re_{\mathrm{I}} = \frac{U_{\mathrm{I}}\varepsilon R_0}{\nu} \qquad (27)$$

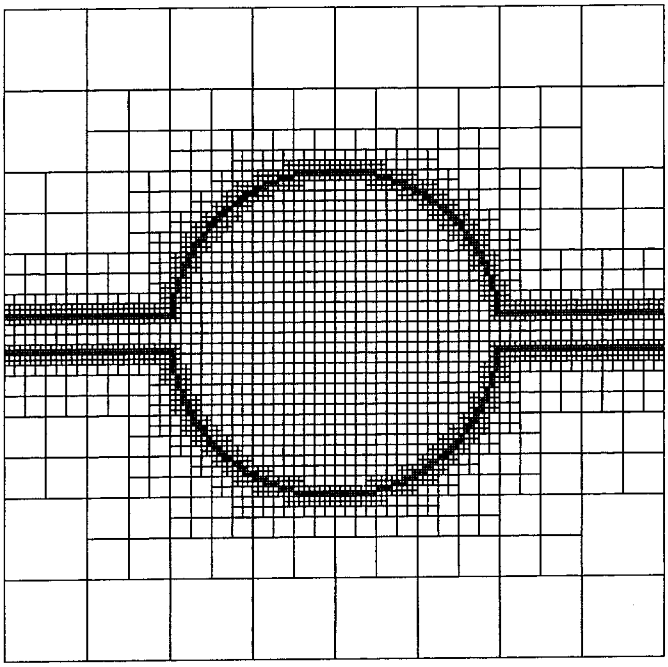

Figure 24. Jet-forced flow in a circular basin: initial Q-tree grid.

where ν is the kinematic eddy viscosity coefficient. Here, ν is $7.84 \times 10^{-4}\ \mathrm{m^2\,s^{-1}}$, corresponding to $Re_{\mathrm{I}} = 10$. The time step is set as $\Delta t = 0.005$ s.

During the time stepping procedure, the Q-tree grid is adapted according to criteria based on the magnitude of the non-dimensionalized depth-averaged vorticity

$$\Omega_0 = \frac{R_0}{U_{\mathrm{I}}}\left|\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right| \tag{28}$$

Grid enrichment is undertaken whereby cells of resolution level less than 8 are sub-divided if $\Omega_0 > 2.25$. Grid coarsening takes place by removal of all four cells, within a given parent cell, if their resolution level is greater than 6 and the maximum value of the four children is $\Omega_0 < 0.75$. Figures 25 and 26 depict the adapted grid and steady state velocity vector field at time $t = 45$ s. As would be expected, the flow separates on entering the basin; there is a bounded through-flow stream between the inlet and outlet stems, which bulges towards the outlet before exiting as a radial sink-like flow. Symmetric vortices are produced in the fluid regions either side of the through-flow, their centres displaced slightly towards the inlet side of the basin. The upper vortex is anticlockwise, the lower clockwise rotating. The flow patterns are almost identical to those produced by other numerical schemes for solving the shallow flow equations (e.g. References [3,30]). The depth-averaged streamlines in Figure 27 are in close

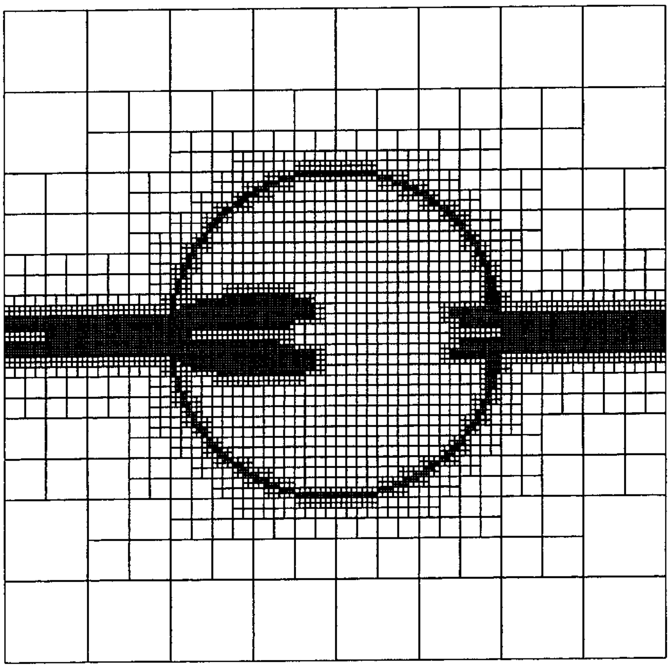

Figure 25. Jet-forced flow in a circular basin: final adapted Q-tree grid, $Re_{\mathrm{I}} = 10$.

agreement with Dennis' [28] two-dimensional results for viscous flow in a circular domain. This test confirms the scheme's ability to model recirculating flow within a geometrically complicated domain.

6. CONCLUSIONS

A second-order accurate Godunov-type solver of the SWEs in finite volume form is described. A new mathematical formulation is derived that accounts properly for spatially varying bathymetry by balancing the flux gradient and source terms, and is suitable for Roe's approximate Riemann scheme without the need for *a posteriori* numerical techniques. The model utilizes Q-tree grids, with adaptation criteria based on the magnitudes of free surface gradient or depth-averaged vorticity. Results are presented for several validation tests, which highlight certain features of the present shallow flow solver. The effect of non-uniform bed conditions is examined by simulating wind-induced steady state circulation in a dish-shaped circular basin. The flow pattern is in broad agreement with Kranenburg's [23] model. However, the centres of the gyres are slightly downwind of the locations assumed by Kranenburg. Moving-front dam-break and shock-type hydraulic jump flows are also considered, and the grid adaptation found to lead to significant array storage savings while retaining

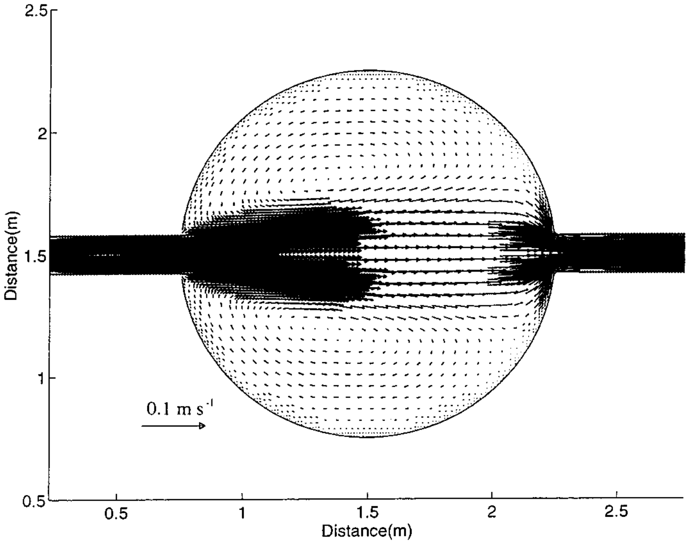

Figure 26. Jet-forced flow in a circular basin: depth-averaged velocities, $Re_I = 10$.

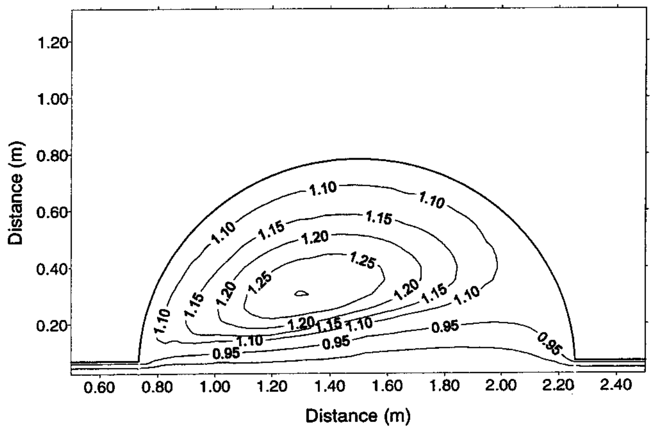

Figure 27. Jet-forced flow in a circular basin: depth-averaged streamlines, $Re_I = 10$.

accuracy. Laminar jet-forced flow in a circular reservoir is computed, a case that involves a relatively complicated geometry and tests advection and diffusion terms in the SWEs. The steady state flow field contains counter-rotating recirculation regions either side of the jet flow between inlet and outlet. Overall, the results are in close agreement with alternative semi-analytical and numerical models given in the literature.

ACKNOWLEDGMENTS

This work was undertaken while the second author was an academic visitor at the University of Oxford on a fellowship (outbound) from the Ministry of Education, Science, Sports and Culture of Japan. The work has been supported by the U.K. Engineering Science and Research Council through EPSRC Grant GR/L92877, and co-investigated by Dr K Anastasiou of Imperial College, London, and Dr P.H. Taylor of Oxford University.

APPENDIX A. NOMENCLATURE

a	$un_x + vn_y$
$\mathbf{A}$	inviscid flux Jacobian matrix
c	wave celerity
f	Coriolis parameter
$\mathbf{f}$, $\mathbf{g}$	flux vectors
$\hat{\mathbf{f}}$	vector of flux functions through S
$\mathbf{f}^{\mathrm{I}}$	vector of inviscid fluxes
$\mathbf{f}^{\mathrm{V}}$	vector of viscous fluxes
g	gravitational acceleration
h	total water depth
h_s	still water depth
$\mathbf{h}$	vector of forcing functions
H	weighted mean depth
L	free surface wavelength
$\mathbf{L}$	left eigenvector matrix
n	$\sqrt{n_x^2 + n_y^2}$
n_x, n_y	Cartesian components of $\mathbf{n}$
$\mathbf{n}$	unit normal vector to S
N_A	identification number of cell A
q	component of vector of conserved variables, $\mathbf{q}$
q_A	value of q at centre of cell A
$\mathbf{q}$	vector of conserved variables
$\mathbf{q}_{i,j}^{+}$, $\mathbf{q}_{i,j}^{-}$	reconstructed right and left Riemann states at interface between cells i and j
r	gradient ratio
r_b	distance from centre of circular basin
$\mathbf{r}$	distance vector from cell centre to any arbitrary point within the cell
R_0	radius of circular basin
$\mathbf{R}$	right eigenvector matrix
Re_{I}	inlet Reynolds number
S	boundary of problem domain
S_{ox}, S_{oy}	bottom slope components in the x- and y-directions respectively
t	time
T_r	time required to build up the flow
u_*	shear velocity at free surface
u, v	depth-averaged velocity components in the x- and y-directions respectively

$\tilde{u}$, $\tilde{v}$	depth-averaged velocity components normal and tangential to boundary
U_{I}	inlet flow velocity
V	cell area
x, y	horizontal Cartesian co-ordinates in physical plane
z_0	bed roughness height
Z	normalized depth, $Z = H/z_0$

Greek letters

β	limiter parameter
β_0	oblique hydraulic jump angle
ε	half-angle defining basin inlet and outlet openings
κ	von Kármán constant
λ_1, λ_2, λ_3	eigenvalues
$\lvert\Lambda\rvert$	diagonal matrix of the absolute values of the eigenvalues of **A**
v	coefficient of kinematic eddy viscosity
θ	angle of converging wall
ρ	density of water
τ_{ss}	magnitude of steady state surface shear stress
τ_w	magnitude of free surface shear stress
τ_{wx}, τ_{wy}	free surface shear stress components in the x- and y-directions respectively
τ_{bx}, τ_{by}	bed shear stress components in the x- and y-directions respectively
ζ	free surface elevation above still water level
Δs	length of cell side
Δt	time step
Δx	square cell dimension
Φ	slope limiter
∇	gradient vector
Θ	free surface gradient adaptivity parameter
Ω	problem domain
Ω_0	vorticity adaptation parameter
∂C	cell boundary

Subscripts

B	outer Riemann state at boundary
E	east
i, j	cell number indices
I	inner Riemann state at boundary
N	north
S	south

x, y	derivatives in the x- and y-directions respectively
W	west

Superscripts

n	time levels
I, V	inviscid and viscous components respectively
$+$, $-$	right and left Riemann states either side of a cell interface

REFERENCES

1. Lin B, Falconer RA. Modelling sediment fluxes in estuarine waters using curvilinear coordinate system. Estuary. *Coastal and Shelf Science* 1995; **14**: 413–428.
2. Borthwick AGL, Akponasa GA. Reservoir flow prediction by contravariant shallow water equations. *Journal of Hydraulic Engineering* 1997; **123**(5): 432–439.
3. Anastasiou K, Chan CT. Solution of the 2D shallow water equations using the finite volume method on unstructured triangular meshes. *International Journal for Numerical Methods in Fluids* 1997; **24**: 1225–1245.
4. Gáspár C, Józsa J, Sarkkula J. Shallow lake modelling using quadtree-based grids. In *Proceedings of the X International Conference on Computational Methods in Water Resources*, Heidelberg, Peters A, *et al*. (eds). Kluwer Academic Publishers: Amsterdam, 1994; 1053–1063.
5. Borthwick AGL, Cruz S. Adaptive hierarchical meshing for localized modelling of shallow flows. In *Proceedings of the Third International Conference on Fluid Mechanics*, Beijing, Yanan Tang, Dagang Zhao (eds). Beijing Institute of Technology Press: Beijing, 1998; 903–908.
6. Greaves DM, Borthwick AGL. On the use of adaptive hierarchical meshes for numerical simulation of separated flows. *International Journal for Numerical Methods in Fluids* 1998; **26**: 303–322.
7. Wille SØ. Adaptive finite element simulations of the surface currents in the North Sea. *Computer Methods in Applied Mechanics and Engineering* 1998; **166**: 379–390.
8. Godunov SK. A difference method for the numerical computation of discontinuous solutions of hydrodynamic equations. *Math Sbornik* 1959; **47**: 271–306, in Russian.
9. Roe PL. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics* 1981; **43**: 357–372.
10. Toro EF. *Riemann Solvers and Numerical Methods for Fluid Dynamics—A Practical Introduction*. Springer: Berlin, 1997.
11. Hirsch H. *Numerical Computation of Internal and External Flows*, *Vol. 2. Computational Methods for Inviscid and Viscous Flows*. Wiley: New York, 1990.
12. Ambrosi D. Approximation of shallow water equations by Roe's Riemann solver. *International Journal for Numerical Methods in Fluids* 1995; **20**: 157–168.
13. Nugic M. Efficient implementation of non-oscillatory schemes for the computation of free-surface flows. *Journal of Hydraulic Research* 1995; **33**(1): 101–111.
14. Vázquez-Cendón ME. Improved treatment of source terms in upwind schemes for the shallow water equations in channels with irregular geometry. *Journal of Computational Physics* 1999; **148**: 497–526.
15. Hubbard M, Garcia-Navarro P. Balancing source terms and flux gradients in finite volume schemes. In *Proceedings of Godunov Methods: Theory and Applications*, Oxford, 18–22 October, Toro EF (ed.). Kluwer Academic Publishers: Amsterdam, 1999.
16. Van Dommelen L, Rundensteiner EA. Fast adaptive summation of point forces in the two-dimensional Poisson equation. *Journal of Computational Physics* 1989; **83**: 126–147.
17. Greaves DM, Borthwick AGL. Hierarchical tree-based finite element mesh generation. *International Journal for Numerical Methods in Fluids* 1999; **45**: 447–471.
18. Falconer RA. An introduction to nearly-horizontal flows. In *Coastal*, *Estuarial and Harbour Engineers Reference Book*, Abbott MB, Price WA (eds). Chapman & Hall: New York, 1993; 27–36.
19. Alcrudo F, Garcia-Navarro P. A high-resolution Godunov-type scheme in finite volumes for the 2D shallow-water equations. *International Journal for Numerical Methods in Fluids* 1993; **16**: 489–505.
20. Zhao DH, Shen HW, Lai JS, Tabios III GQ. Approximate Riemann solvers in FVM for 2D hydraulic shock wave modeling. *Journal of Hydraulic Engineering*, *ASCE* 1996; **122**(12): 692–702.

21. Van der Maarel HTM. A local grid refinement method for the Euler equations. PhD thesis, University of Amsterdam, 1993.
22. Cruz S. Numerical solution of the shallow water equations on quadtree grids. D.Phil. thesis, University of Oxford, 1997.
23. Kranenburg C. Wind-driven chaotic advection in a shallow model lake. *Journal of Hydraulic Research* 1992; **30**(1): 29–46.
24. Zhao DH, Shen HW, Tabios III GQ, Lai JS, Tan WY. Finite-volume two-dimensional unsteady-flow model for river basins. *Journal of Hydraulic Engineering*, *ASCE* 1994; **120**(7): 863–883.
25. Louaked M, Hanich L. TVD scheme for the shallow water equations. *Journal of Hydraulic Research* 1998; **36**(3): 363–378.
26. Mingham CG, Causon DM. High-resolution finite-volume method for shallow water flows. *Journal of Hydraulic Engineering*, *ASCE* 1998; **124**(6): 605–614.
27. Hager WH, Schwalt M, Jimenez O, Chaudhry MH. Supercritical flow near an abrupt wall deflection. *Journal of Hydraulic Research* 1994; **32**(1): 103–118.
28. Dennis SCR. Application of the series truncation method to two-dimensional flows. In *Proceedings Fourth International Conference on Numerical Methods in Fluid Dynamics*, Richtmyer RD (ed.). Springer: New York, 1974; 146–151.
29. Mills RD. Computing internal viscous flow problems for the circle by integral methods. *Journal of Fluid Mechanics* 1977; **79**(3): 609–624.
30. Borthwick AGL, Barber RW. River and reservoir modelling using the transformed shallow water equations. *International Journal for Numerical Methods in Fluids* 1992; **14**: 1193–1217.